

Information Flows at Inter-Team Boundaries in Agile Information Systems Development

Scarlet Rahy^{1*} and Julian Bass²

1 University of Salford, School of Computing, Science & Engineering, Manchester, UK
S.Rahy@edu.salford.ac.uk

2 University of Salford, School of Computing, Science & Engineering, Manchester, UK
J.Bass@salford.ac.uk

Abstract. *Agile software development methods are being used on larger projects thus the study of inter-team communication are becoming an important topic of interest for researchers. This research addresses inter-team communication by exploring the tools and three different boundaries, inter-team, team and customers, and geographically separated teams. In this research, we gathered data from semi-structured face-to-face interviews which were analyzed following the grounded theory approach. Our study reveals consensus from different teams on the importance of virtual Kanban boards. Also, some teams members tend to adapt to other teams' preferred communication tool. We observed challenges around interdependent user stories among the different teams and highlighted the problems that rise at the different boundaries.*

Keywords: agile information system development • inter-team communication • agile team boundary • communication • agile methods • cooperating agile teams

1. Introduction

Since the creation of the agile manifesto in 2001, agile methods enhanced the customer involvement, adaptability, and evolutionary delivery in software development [11]. Agile software development is growing horizontally within different organizations and deeply within the same organization [9]. Globalization is irreversible and so is the nature of the software development industry as a digital currency [15]. Multi-cultural and geographically distributed software development models are becoming more common. Thus rises the need to study how agile software development teams can work effectively in these geographically distributed settings [1].

Inter team communication is a crucial part to achieve the success of agile software development [10, 17, 25, 26]. Agile teams cannot work in isolation and thus coordination is a necessity. This study examines the means of communication used across the different teams through observing a case of a software development company spreading across two geographical locations, The Netherlands and Kenya. This study also addresses the issue of communication across three different boundaries, inter-teams, teams and customer, and geographically separated teams.

This paper is structured as follows. In the next section, an overview of the literature review is presented and includes an overview of agile and inter-team information flow in agile. Then the research method adopted is introduced providing information on the research site, data collection and data analysis. Then findings are presented and divided into main section, inter-team communication tools and communication at the boundaries. Finally at the end there is the discussion, recommendation, and conclusion.

2. Related Work

2.1 Overview of agile

The Agile Manifesto has organized and made clear the application of agile in the software development industry. The agile methods are based on the values and philosophies developed in the Agile Manifesto [6]. It promotes people and social focused views on software development. The goal of agile application lies in adaptability, flexibility and responsiveness [14]. Agile adapts to the constantly changing word by learning through experimentation and introspection and adopting it as a problem solving method [14]. There are several agile methods that are applied in software development such as Lean Software Development [23], Scrum, and Extreme Programming.

Inter-team communication is highly related to the agile practices used. Furthermore, there are several agile practices that are adopted including Refactoring, Release Planning, Velocity, Iteration Planning, and Coding Standard [8]. The most used agile practice is the Daily Stand-up meetings, with 90% usage, followed by Sprint Planning, with 88% usage, and ranking third are the Retrospectives, with 85% usage [9] all of which demand effective communication skills. The Daily Stand up Meetings occur daily between the team members in a prearranged space and time to discuss what has been done, what is going to be done and impediments encountered, if any [27]. Sprint planning occurs when team members gather to share the details on user stories' complexity, utility, and dependency [7]. Retrospectives are devoted for the improvement of the agile software development process and for adaptation to changes that arise [18].

2.2 Inter-Team Information Flow in Agile Software Development

Agile software development is based on inter-team collaboration and coordination [19, 25]. These teams and their members are known for the dynamic behavior that is able to adjust according to the customer's requirements; for customers are major influencers in the agile software development process [19, 24]. The identification and prioritization of customer requirements is conflicting in agile software development [5, 24]. Some team members do not accept criticism and perceive it as a personal offense and subsequently retreat and defend themselves rather than their ideas or work [17].

Agile team members are meant to be democratic, all team members are equal [17, 20]. Moreover, the Agile Manifesto guarantees that all team members have equal opportunity in the decision making process [6]. This enhances the self-organizing ability of teams and introduces it as a mean to achieve the best design, architecture, and customer requirements [6, 17]. Self-organizing teams in agile are characterized by communication, feedback, coordination, and collaboration [13]. But this collaborative nature introduces obstacles that team members face in relation to the decision making principles. A major obstacle is the unwillingness of team members to commit to a decision. Rather, team members tend to consider decision making as a burden rather than a privilege and rely on the Scrum Master for decision taking [13]. In these cases, Scrum Masters tend to choose either to take the decision and inform the team members, thus violating an agile principle, encourage team members and wait for their response, or use decision making support systems to aid in the process [4].

Effective inter-team knowledge sharing is highly important in agile software development. Santos and Goldman developed a theoretical model for inter-team knowledge sharing effectiveness. This model highlighted two influencing factors that are the organizational conditions and stimuli. Organizational conditions are identified as top management, team integration, environment and

agile methods adopted. While stimuli are motivators that include common goals and incentives [25]. Dingsoyr et al. (2018) found a balance between the centralized behavior and the self-management agile driven behavior. Smite et al. (2017) highlighted the importance of networking and cultivating teams to practice cross team interaction. Also, boundary spanners act as coordinators who provide a source of information, a target for feedback [26], a mediator between different teams [29], and a sociomaterial assemblages [12].

There is extensive research on agile software development principles and practices [2, 3, 5, 16, 22] but less research is done on inter-team communication. Inter-team knowledge sharing in agile software development are still in the rise [10, 25, 26]. Inter-team communication is identified as an important topic in research [10]. Practices that are applied specifically for knowledge sharing in agile software development are still under study. Our study has added to the research in inter-team communication tools used and problems that rise at the boundaries between the different teams.

3. Method

The qualitative research methodology is adopted by this study. Specifically, qualitative research is used as a basis for implementing the grounded theory approach. Grounded theory was used since it enabled the suspension of preconceptions and the analysis of new concepts from the data. The three pillars of qualitative research include open ended interviews, direct observation, and written communication [21]. In this study, data was collected using semi-structured open ended interviews. The unit of analysis are employees [28] and product owners at this study's research site. An exploratory pilot study was conducted in the first phase to enable refinement and possible adjustments in the questions along with the familiarization with this type of research. In the second phase, a deductive synthesis of a series of interviews was done to enable analysis.

3.1 Research Sites

Data was obtained and analyzed from an international company providing services in software development. The company develops software using agile software development techniques and conducts administrative work also using agile. The international company has a main office in The Hague, Netherlands, 50 employees, and a partner office in Nairobi, Kenya, 15 employees. The company was chosen according to the snowball sampling technique; academic contact eased the connection. In the second phase, the professional contact provided access to study participants.

3.2 Data collection

Participants were interviewed with different responsibilities and locations in the company. An overview of the participants' location, role, and responsibility are shown in Table 1. The data collected was obtained from semi-structured open-ended questions. All interviews were recorded after obtaining the practitioners' consent. Then interviews were transcribed manually since it ensures correct transcription and reminds the interviewer of the social and emotional aspects that occurred during the interview [28]. The most effective way to optimize the data collected from interviews is to record and transcribe data manually [2]. The conducted interviews followed a guide open-ended questions that enabled the participants to raise any issue that came up even if it wasn't mentioned in the guide.

Table 1: Participants' Roles, Responsibility, and Location

Participant	Role and Responsibility	Location
Director-1	Director of the Company	Kenya
Director-2	Director of the Company and Product Owner	Netherlands / Kenya
Technical Lead	Technical Lead and Product Owner	Netherlands / Kenya
Public Relations (PR)	Public Relations Manager	Netherland
Sales	Sales Coordinator	Netherlands
Designer	Designer and part of the Public Relations team	Netherlands
Developer-1	Scrum Master and Developer	Kenya
Developer-2	Front End Developer	Netherlands
Human Resource (HR)	Human Resources Manager	Netherlands

3.3 Data analysis

The transcribed data was imported to an analyzing tool Nvivo 11. All interviews were coded, leading to deriving categories, high levels of abstraction, and concepts and patterns of behavior [2]. The categories were deduced solely from the transcribed interviews without inducing any preconceived ideas or thoughts. Significant points were highlighted from each interview and then compared with other interviews. This constant comparison technique was a key to identify concepts that were then grouped into categories that were coded.

Line-by-line open coding approach was used on the transcribed interviews. When coding line-by-line, data can be inspected and a special incident can be found in a word, a line, or through several lines [2]. This coding process was organized using the Nodes option in Nvivo. Each code was given a title and constant comparison method was used. The transcribed interviews were reviewed more than once and each time new categories emerged. This ensured that no data was left unnoticed. This process stopped when no new categories were created and theoretical saturation was reached.

The next step was the writing of the memos. After writing each memo roughly, the memos were revisited and written in a formal. This will ensure that memos are written in the “passion of the movement” [2] and guarantee that memos will be understood in the correct way through using correct and revised English. Also, constant comparison was applied to categories and the participants’ responses from the two different geographical locations were compared. Thus, when a difference in opinion occurred, if any, it was indicated in the writing of the memos. Quotes from the interviews were used as evidence in the writing of memos.

4. Findings

The findings in this study are organized into two main parts. The first part discusses the inter-team tools that are used for communication such as team messaging tools, face-to-face, and virtual Kanban boards. The second part discusses the communication boundaries between teams inside the company, between teams and customers, and between teams that are located in separate geographical locations.

4.1 Inter-team Communication Tools

There are several forms of communication that are used to transfer information from one actor to another. These include team messaging tools, face-to-face, emails, virtual Kanban boards...

Team Messaging Tools. Team members use a messaging tool called Slack to chat inside the company. Slack is an application that works as a digital workspace for communication between the different members of the company.

During the interviews, we noticed a major difference in the point of views when it came to the usage of slack. The advocates and daily users of slack were mainly the developers; while the criticizers were the designers, sales, human resources and public relations team.

To begin with, the developers use slack daily as a way to communicate with the team members and other employees. Participants highlighted the different benefits of slack. The first benefit is that slack allows open communication between members of the same team, project, or the company as a whole. According to Developer-1:

“In slack we are able to communicate to a group. A group can keep up with the communication and know what is going on in different aspects. Also they can chip in if they feel there is something they can input on any matter.”

Also the Technical Lead highlighted the benefits of slack as a platform to ask for help when needed: *“...we have channels for the teams on their own and we have channels for the project we have a channel were people can ask for help”*.

Second, slack has features, notifications, tags, and pins, which aid in simplifying and facilitating the communication process. Developer-1 said: *“If we have a group that is specific for a certain project I am able to tag the members of the team to draw their attention to something... everyone will get notified as soon as I post something”*. Moreover, Developer-2 revealed how slack can be used as an updating tool when an employee is absent for a period of time: *“If we need to remember something for the next day we pin it on slack so we can see it the next day”*, and the Technical Lead agreed.

Third, developers use slack for archiving documents and saving conversation. Developer-2 indicated how slack can be used as a memory box: *“Everything that needs to be documented so we don't forget we just put it in [slack]”*.

On the other hand, some other employees weren't that enthusiastic of slack and didn't use it that often. The non-developers used slack solemnly when communicating with the developers. The Designer said: *“I use slack in order to communicate but I mostly do that with developers”*. The Sales person showed displeasure with the usage of slack: *“I use slack but not a lot...the only people I know who use slack are developers”*. The Technical Lead said: *“It is more difficult for non-developers to actually express by text what they mean”*. The dependence on verbal communication and neglecting slack highlights a tradeoff between the benefits of verbal communication and the availability of information for all the involved members. When information is transferred verbally between two members or more, it won't necessary reach all the involved members.

In addition, slack can be used to ask and answer small questions. For instance if a developer has a specific question about a certain color in the design or font this could be best transferred through slack. The Designer said: *"They [developers] always have short questions, like what is the color of this design, how big this should be... and many questions like that during the development process"*.

Face-to-face. Face-to-face communication is a mean to enhance collaboration and creativity in the workplace. When communicating face-to-face the information is not only limited to the words said, it also reveals the body language, tone, reactions and feelings. All the employees use face-to-face communication but some prefer it more than others. Director-2 said: *"When most people use a document we try to use an email. When most people use email we try to use chat message. When most people chat we try to talk to each other"*. Face-to-face communication is used during daily standups, demos, retrospectives, and regularly during the day.

The PR, Sales, HR and designers tend to use face-to-face communication very often. This is their preferred form of communication. The Designer indicated that face-to-face communication is the first step toward creating the design of a certain project: *"Without face-to-face communication I cannot start with my design"*; the HR manager seconds that. Director-1 indicated the importance of body language in communication: *"I prefer face-to-face physical communication. Because communication is not just text it is also body language, tone facial expression which I think you lose most of it if you only type, text or email"*.

One of the main challenges when communicating face-to-face rises when using Skype. For instance, when communicating between The Netherlands and Kenya, the main problem is the poor internet connection. The Designer indicated that *"sometimes in Kenya they are a bit slower and they have trouble with internet connection a lot. Sometimes the sprint would be affected by that"*; Developer-2 also indicated that.

Virtual Kanban Board. The uses of virtual Kanban boards are essential for software development companies implementing agile. The importance of virtual Kanban boards, in this case Trello boards, was recognized by all employees. Trello boards are used to keep track of everyone's work and daily activities. The Technical Lead stressed that Trello boards facilitates the managing process and helps managers and team leaders acquire a general overview of the work. The Technical Lead said: *"Trello keeps an overview of all the project, sprints, and teams at the same time"*.

Moreover, Trello boards allows all members of the organization to check the status of work of different teams by simply examining the board of each team. Trello boards highlights the user stories' status. The Technical Lead said: *"we use Trello and this is where we set all the user stories for the teams....when you pass by you can see straight away what the status is of the sprint"*.

4.2 Communication at the Boundaries

It has been discussed earlier how each person or team prefers a certain type of communication method. This along with other factors cause communication problems at the boundaries between teams, directors, product owners, and customers.

Inter-team Boundaries. One of the main issues in inter-team communication occurs when members do not respond to requests from other teams. Members tend to prioritize the tasks given by their own team leader or scrum master, and postpone the tasks or requests given to them by other team's leaders. The Sales describes an incident that occurred with one of the developers under that context:

"She started working at the customer but the screening wasn't completed yet and she needed to hand a document. I emailed her, called and she wasn't responding for several days and that was really frustrating. It took three days to complete that".

Another issue that rises is how a team tends to assume that other teams know their status and if they are facing any setbacks. Developer-2 said: *"Teams are like, we know so the rest knows it as well. They are just assuming"*. This brings up the role of the scrum master in inter-team communication. The scrum master is responsible to receive and send information to members in and outside his/her team. Developer-2 describes an incident where one team did not finish their sprint and another team was depending on the successful completion of that team's sprint to initiate their own. Developer said: *"If the scrum master had told the other team they could have done a new sprint planning. Now they were just waiting and two teams were set back by that"*.

Absence of Inter-team Communication In The Presence Of Dependencies. During a project, several sprints are interconnected. A sprint for Team A may be a prerequisite for the completion of the Team B's sprint. This reliability sometimes may cause problems especially if the work done by Team A's sprint was not complete on time or needed rework. Developer-2 explained a similar case:

"In the last sprint there was a team who picked up a bit too much, they under estimated their user stories and another team was dependent on what they were supposed to make but they weren't informed of the delay. So they were waiting for their sprint to start for some time."

This case has occurred several times between the design team and the developing team. Since the developing team is highly dependent on the work of the design team and since the sprint of the design team are highly dependent on the customer's approval, the developing team has experienced incidences where they had idle time. The Designer described: *"If something goes wrong in my sprint then the developers don't have anything to do"*.

The Technical Lead suggested that: *"The delay can be caused by miscommunication"*. The lack of communication between teams can cause the delay of sprints and thus creating an idle time. This wait time could have been avoided if communication had occurred. Developer-2 said: *"So if they had told them [about the unfinished work] they could have done a new sprint planning and plan different stuff and finish other tasks"*.

Team and Customer Boundaries. Communication with customers through agencies, dealing with unclear customers' requirements, and reaction to customers' feedback are issues that rise at the team and customer boundary.

Communicating with Customers through Agencies. In an agile software development company, customer feedback directly reaches the employees. In cases where customers are contacted through agencies, the communication is faced with breakdowns. The Technical Lead said: *“when the development is through an agency, the communication is more difficult”*. First, These breakdowns can cause frustration for the employees especially with the involvement of several actors. Second, transfer of false information may occur. Third, employees will feel they have restrictions or limitations while performing their job which may lead to demotivation. The Technical Lead said: *“This becomes time consuming. This is something that we don’t have control over. It is somehow frustrating”*.

Unclear Customer Requirement. Understanding the customer needs and building on their requests are of high importance. Lack of information and clarification may also lead to delay in delivery with respect to time. The PR manager said: *“Also the lack of information of the project we are doing is a factor that will negatively affect delivery time”*. Team members should be willing to ask questions at the beginning of the project before writing the user stories and planning the sprints. The Designer said: *“I ask a lot of questions make the story clear and make a proper sprint planning out of it”*. Director-1 highlights the importance of the scrum master’s role in clarifying the customer’s requirements: *“The scrum master can clarify the story or feature with the customer and product owner and communicate that to the team”*.

Unfortunately, sometimes the team members or the scrum master do not ask the correct questions and base the user stories and sprint planning solemnly on the customer’s briefing. Director-2 said: *“I think the most common negative effect on the workload is the lack of clarity and understanding in what is required”*; the designer also agreed to that. Also some clients might not be involved in the process. Director-1 said: *“The clients give the requirements but they don’t get involved so much”*.

Reaction to Customer’s Feedback. The customer feedback presented in the demo will directly affect the retrospective. Sometimes employees tend to express their feelings about the demo solemnly instead of expressing their feedback about the whole sprint and the demo. Director-2 indicated: *“In our last demo we had problem with the customer that resulted in negative feedback and then in the retro everybody put negative improvement stickys related to that particular incident”*. Some of the employees become directly affected by the customer feedback presented in the demo and it stretches further to affect not only the retrospective but also the performance in the coming day. The Technical Lead indicates that: *“It depends if they get affected with the customer’s feedback”*. On one hand, some employees get affected by the feedback. As the Designer said: *“It [customer feedback] can discourage you especially. I think as a designer you know the feedback will always come up and it is never the way you thought it would be”*. Also Developer-2 talked about a recovery period: *“Usually the demo is on a Thursday and Friday is usually is a personal sprint day and Monday you can just start over. Friday would be like a day to recover”*.

On the other hand, some employees do not get affected by the feedback and understand that this is part of the job especially in the software development industry where clients tend to change their

requirements frequently. Developer-1 indicated: *The scrum agile process is actually about making the software based on the customer's demand*".

Geographically Separated Inter-Team Boundary. During the first steps of agile implementation in Kenya, the Netherlands had a lot to offer and agile implementation was a challenge. The Kenyans were learning the agile process from the Dutch. Developer-1, based in Kenya, said: *"They were able to catch our weaknesses and give us a scrum agile implementation of a solution for the problems"*. But all throughout the implementation process, the people at the Netherlands, learned and enhanced the agile process. The Technical Lead said: *"After every sprint in Kenya we learned new things and we restarted the game"*.

First of all, the communication between the people in Netherlands improved after practicing long-distance communication with the Kenyans. Developer-2, having direct contact with the Kenyans, expressed:

"If you managed to communicate with people in Kenya every day for 10 sprints in a row then it is easier and becomes second nature to talk with people around you as well. Instead of calling them kilometers away you just walk the 5 meters to the next room and you talk with them".

Other employees, who were not in contact with the Kenyans, saw this improvements in their colleagues and started enhancing their communication skills. The Technical Lead said: *"The whole company learned a lot about communication especially in the least two weeks...our communication improved immensely"*; Director-2 expressed the same.

Second, the Kenyans found a new way to design the Trello board and the office back in the Netherlands adopted this change. The Technical Lead said:

"In Kenya they came up with some ways to organize their Trello board for their office team, management team. And we adapted that and then we changed that for a more suitable way for our office".

Third, the way user stories were written in the Netherlands also changed. After implementing agile in the Kenya, the need for detailed and descriptive user stories emerged. The Technical Lead explained:

"Because we added extra description in Kenya, we found out that we need to give a little more information that they can look up afterwards instead of telling that in the sprint planning next to the user story".

5. Discussion

This study responds to the call for further research on inter-team communication and customer involvement [10]. Thus our paper offers a case study that looks into inter-team communication, the methods used, and the difficulties that rise at the boundaries between different actors, employees and customers, and across diverse locations. In addition, our study reveals a combination of technology choices, organizational boundaries, and interfaces. Both aspects turned out to be a challenge in inter-team communication.

First, our findings show that employees fell into two groups when it came to inter-team communication means and preference. Developers, on one hand, preferred slack and recognized the importance of Trello boards and face-to-face. On the other hand, designers, sales, HR, and PR preferred face-to face communication while recognizing the importance of Trello boards and using Slack when communicating with developers. This reveals how people tend to adapt and improvise ways in order to enhance communication.

There was consensus among the participants in our reproach study that using virtual Kanban board is appropriate. Virtual Kanban boards are means to keep all employees up-to-date while highlighting the user stories without disturbing the flow of work. We observe that developers in this study are already benefiting from slack's features, tagging, pinning, and notifications while endorsing its usage in specific cases such as short clarifications, information storing platform when an employee is unavailable, and public information broadcasting. It is surprising that HR, PR, sales and designers are not seeing the benefits of slack but rather using it as means to communicate with developers. Maybe directors would see benefit in improving communication if they encourage face-to-face information sharing and networking behavior through offering communication skills training [26], giving cross team effective feedback, and encouraging ad-hoc conversations. 55/38/7 rule stresses on the importance of body language and tone and these can only be portrayed through face-to-face communication.

Second, we studied communication at the boundaries at three different levels: inter-team boundary, team and customer boundary, and geographically separated inter-team boundary. Previous literature has shown six informal roles in self-organizing agile teams that will smoothen the communication at the boundaries [17]. Our study reveals some of the problems at the boundaries while highlighting observations to enhance communication at the different boundaries. The figures below highlight the bottlenecks experienced and their respective results at the inter-team and geographically separated inter-team boundaries (figure 1) and team customer boundary (figure 2).

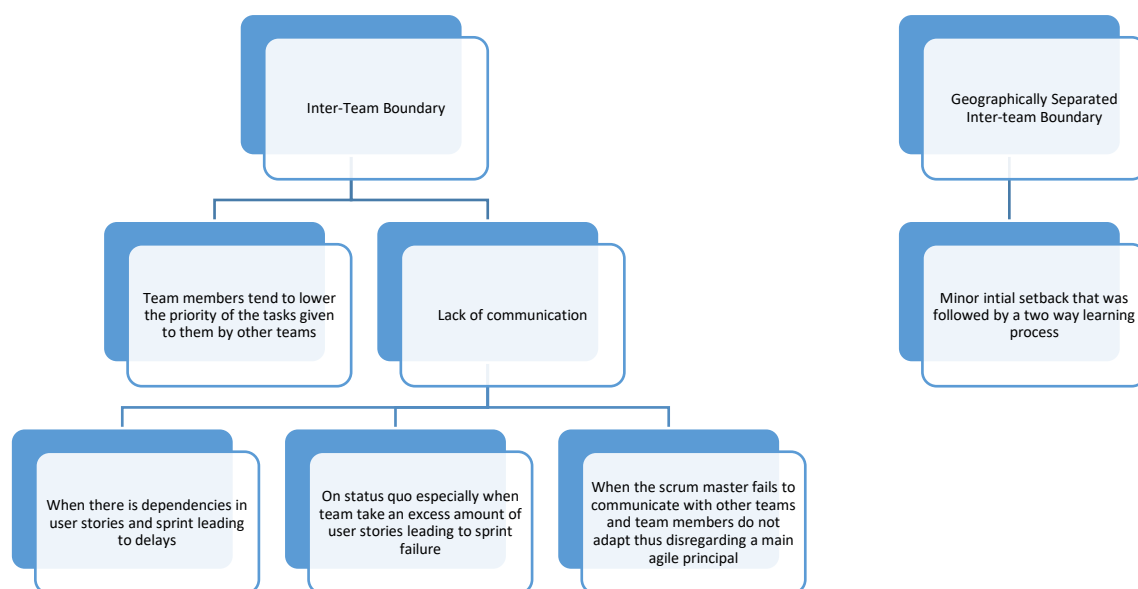


Figure 1: Bottlenecks at inter-team and geographically separated inter-team boundaries

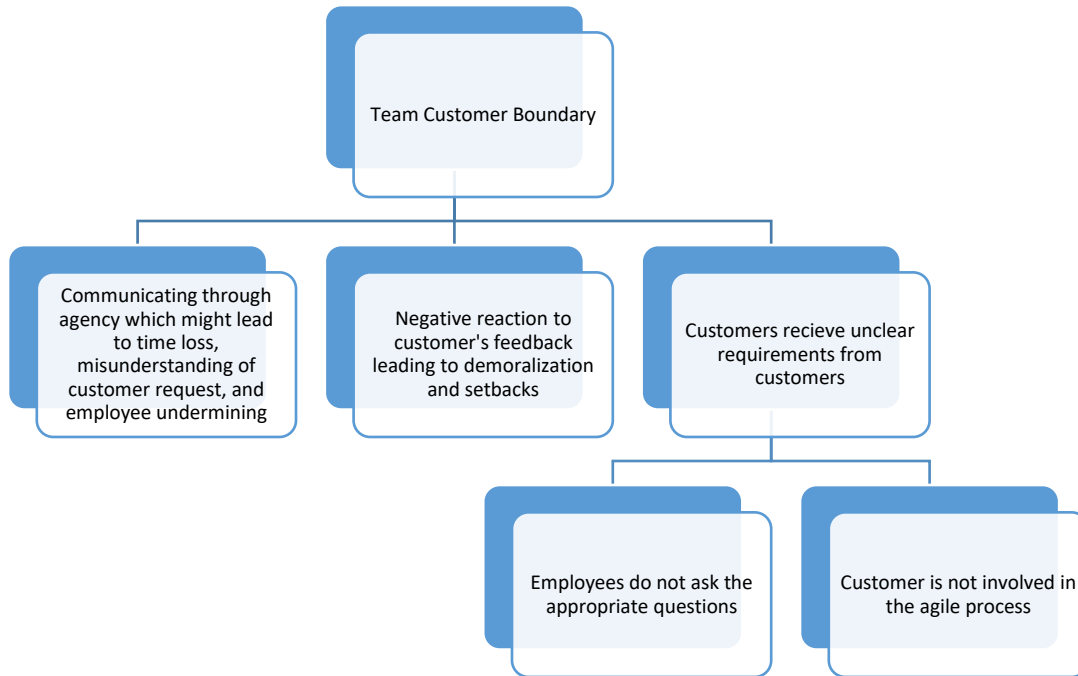


Figure 2: Bottlenecks at team customer boundary

Smooth communication at the boundaries is hard to achieve [12, 29]. At the level of inter-team boundary, in addition to applying the communication tools mix provided earlier, team members may consider enhancing communication especially when user stories are interconnected, dependent, or are on the verge of failure. For example, this could be done by highlighting dependent user stories on the Trello board and ensuring the scrum masters involved conduct ad-hoc meeting to discuss updates. But if the scrum master failed to do so this may trigger the team members to jump in and perform the communication since in agile everyone is considered equal; and at the retrospective this can be highlighted, discussed, and resolved.

At the level of team and customer boundary, employees should understand that customer involvement and feedback is in the core of agile software development. Thus negative feedback should neither demoralize nor discourage employees but rather it should encourage them. This encouragement may come from the product owner especially while conducting the retrospective that should include constructive criticism. In addition, employees should learn to ask specific questions to the customers especially when the requirements are vague. Further research may involve creating general guidelines for customer requirement clarification. Moreover, the role of boundary spanners should be supported and enhanced.

At the level of the geographically separated inter team boundaries, unfamiliar tasks, lack of product knowledge, and cultural differences enhance communication and motivate particular networking behavior [26]. In addition, we observed that team members, having different levels of experience, exchange ideas leading to mutual benefits and learnings.

6. Conclusions

This study used the grounded theory approach to analyze inter-team communication tools at different boundaries. The study was focused on an agile software development company located in The Netherlands and Kenya and interviewed practitioners from different teams and later performed line-by-line coding, memoing, and constant comparison to the data to obtain two main findings. The first revealed the different tools used in inter-team communication and the difference in perceptions between practitioners about the three main tools, Slack, Trello, and face-to-face. The second revealed the problems faced at the levels of the boundaries, inter-teams, teams and customer, and geographically distributed teams. The contribution of this paper is to observe that communication has improved when people deliberately adopt somebody else's preferred communication mechanism.

We discovered different mechanisms that were used at the different boundaries. Teams learn how to benefit from the best of each type of communication tool and how to adapt to certain tools in order to enhance communication. Interdependent user stories may be highlighted on virtual Kanban boards and ensure information exchange about the status quo is achieved through encouraging ad-hoc meetings between scrum masters. In addition, boundary spanners play an important role in reducing the gap between different actors. Finally, geographically separated teams may use their differences in their favor in order to benefit and learn from each other.

References

- [1] Aebert, C., 2012. *Global software and IT; a guide to distributed development, projects, and outsourcing*. Portland: Ringgold Inc.
- [2] ADOLPH, S., HALL, W. and KRUCHTEN, P., 2011. Using grounded theory to study the experience of software development. *Empirical Software Engineering*, **16**(4), pp. 487-513.
- [3] ALAHYARI, H., BERNTSSON SVENSSON, R. and GORSCHER, T., 2017. *A study of value in agile software development organizations*.
- [4] ANIKA RANI, SHAHPER VODANOVISH, DAVID SUNDARAM, 2015. Ubiquitous Decision Making and Support: A Framework and Evaluation, *European, Mediterranean & Middle Eastern Conference on Information Systems 2015*, June 1st-2nd 2015.
- [5] BASS, J.M., 2015. How product owner teams scale agile methods to large distributed enterprises. *Empirical Software Engineering*, **20**(6), pp. 1525-1557.
- [6] BECK, K., BEEDLE, M., VAN BENNEKUM, A., COCKBURN, A., CUNNINGHAM, W., FOWLER, M., GRENNING, J., HIGHSMITH, J., HUNT, A. and JEFFRIES, R., 2001. Manifesto for agile software development.

- [7] BOSCHETTI, M.A., GOLFARELLI, M., RIZZI, S. and TURRICCHIA, E., 2014. A Lagrangian heuristic for sprint planning in agile software development. *Computers and Operations Research*, **43**, pp. 116-128.
- [8] CAMPANELLI, A.S. and PARREIRAS, F.S., 2015. Agile methods tailoring – A systematic literature review. *The Journal of Systems & Software*, **110**, pp. 85-100.
- [9] COLLABNET, A. and VERSION ONE, A., 2018. *The 12th Annual State of Agile Report: Long Term Survey*. Atlanta: VersionOne.
- [10] DINGSØYR, T., MOE, N., FÆGRI, T. and SEIM, E., 2018. Exploring software development at the very large-scale: a revelatory case study and research agenda for agile method adaptation. *Empirical Software Engineering*, **23**(1), pp. 490-520.
- [11] DINGSØYR, T., NERUR, S., BALIJEPALLY, V. and MOE, N.B., 2012. *A decade of agile methodologies: Towards explaining agile software development*.
- [12] DOOLIN, B., 2012. Sociomateriality and boundary objects in information systems development. *European Journal of Information Systems*, **21**(5), pp. 570-586.
- [13] DRURY, M., CONBOY, K. and POWER, K., 2012. Obstacles to decision making in Agile software development teams. *The Journal of Systems & Software*, **85**(6), pp. 1239-1254.
- [14] DYBA, T. and DINGSOYR, T., 2009. What Do We Know about Agile Software Development? *Software, IEEE*, **26**(5),.
- [15] HERBSLEB, J.D. and MOITRA, D., 2001. Global software development. *Software, IEEE*, **18**(2), pp. 16-20.
- [16] HODA, R., NOBLE, J. and MARSHALL, S., 2012. Developing a grounded theory to explain the practices of self-organizing Agile teams. *Empirical Software Engineering*, **17**(6), pp. 609-639.
- [17] HODA, R., NOBLE, J. and MARSHALL, S., 2013. Self-Organizing Roles on Agile Software Development Teams. *Software Engineering, IEEE Transactions on*, **39**(3), pp. 422-444.
- [18] JOVANOVIĆ, M., MESQUIDA, A.-. and MAS, A., 2015. *Process improvement with retrospective gaming in agile software development*.
- [19] LINDVALL, M., BASILI, V., BOEHM, B., COSTA, P., DANGLE, K., SHULL, F., TESORIERO, R., WILLIAMS, L. and ZELKOWITZ, M., 2002. Empirical findings in agile methods, *Conference on extreme programming and agile methods 2002*, Springer, pp. 197-207.
- [20] NERUR, S., MAHAPATRA, R. and MANGALARAJ, G., 2005. Challenges of migrating to agile methodologies. *Communications of the ACM*, **48**(5), pp. 72-78.
- [21] PATTON, M.Q., 1945-, 2002. *Qualitative research and evaluation methods*. 3rd ed. edn. London: London : SAGE.
- [22] PETERSEN, K., ROOS, P., NYSTRÅM, S. and RUNESON, P., 2014. Early identification of bottlenecks in very large scale system of systems software development. *Journal of Software: Evolution and Process*, **26**(12), pp. 1150-1171.
- [23] POPPENDIECK, M.(B.), 2003. *Lean software development : an agile toolkit*. Boston Mass.] ; London: Boston Mass. ; London : Addison-Wesley.

- [24] PRABY, R. and ROLAND, W., 2016. Can Agile Processes Prevent Spectacular Information Systems Development Failures, *European, Mediterranean & Middle Eastern Conference on Information Systems EMCIS 2016*, June 23rd-24th 2016 2016.
- [25] SANTOS, V., GOLDMAN, A. and DE SOUZA, C., 2015. Fostering effective inter-team knowledge sharing in agile software development. *Empirical Software Engineering*, **20**(4), pp. 1006-1051.
- [26] ŠMITE, D., MOE, N.B., ŠABLIS, A. and WOHLIN, C., 2017. Software teams and their knowledge networks in large-scale software development. *Information and Software Technology*, **86**, pp. 71-86.
- [27] STRAY, V., SJØBERG, D.I.K. and DYBÅ, T., 2016. The daily stand-up meeting: A grounded theory study. *The Journal of Systems & Software*, **114**, pp. 101-124.
- [28] VAIVIO, J., 2012. *Interviews – Learning the Craft of Qualitative Research Interviewing*. Routledge.
- [29] YOO, Y., 2008. The dynamics of IT boundary objects, information infrastructures, and organisational identities: the introduction of 3D modelling technologies into the architecture, engineering, and construction industry. *European Journal of Information Systems*, **17**(3), pp. 290-304.